

2

NTATION PAGE

Form Approved
OMB No. 0704-0188

AD-A219 919

It takes an average of 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering the collection of information. Send comments regarding this burden estimate or any other aspect of this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

RT DATE

3. REPORT TYPE AND DATES COVERED

Final Report, 01 Jun 87 thru 31 May 88

4. TITLE AND SUBTITLE

INTEGRATED INCREMENTAL CASE-BASED UNDERSTANDING
AND EXPLANATION: DMAP

5. FUNDING NUMBERS

AFOSR-87-0295
61102F 2304/A7

6. AUTHOR(S)

Professor Roger C. Schank

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Yale University
Dept of Computer Science
Arthur K. Watson Hall
51 Prospect St.
New Haven, CT 06520

8. PERFORMING ORGANIZATION
REPORT NUMBER

90-0351

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NM
Bldg 410
Bolling AFB DC 20332-6448

10. SPONSORING/MONITORING
AGENCY REPORT NUMBER

AFOSR-87-0295

11. SUPPLEMENTARY NOTES

DTIC
ELECTE
MAR 29 1990
S E D

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

We have the following goals for our research with the direct memory access algorithm for understanding and inference: a. using the DMA algorithm to carry out larger scale case-based reasoning, b. improving the robustness of the understander, c. exploring issues in parallelizing the algorithm. In what follows below, we will not make detailed reference to the algorithm, except to distinguish the two basic components of memory search - a. concept refinement, which goes from an abstract memory structure and some components of it to the most specific version of that memory structure that contains those components, and b. concept reference, which uses concept sequences to go from references to certain component concepts to the larger concept that contains those concepts. Concept refinement, for example, goes from "a communication event by Milton Friedman" to "Milton Friedman's argument about interest rates." Concept reference goes from "interest rates", and "soar" to the concept of rising interest rates.

14. SUBJECT TERMS

15. NUMBER OF PAGES

12

16. PRICE CODE

17. SECURITY CLASSIFICATION
OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION
OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION
OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF ABSTRACT

SAR

1 Progress During the Current Period (6 /87 - 5/88)

Work in the current period has continued in two primary areas:

- *Economic Reasoning: DMAP* — Chris Riesbeck, research scientist, and Charles Martin.
- *Case-based planning* — continuing work in the vein of Kris Hammond's CHEF program, conducted by Eric Jones.

Jones and Martin are advanced graduate students. Their work will be continued by additional graduate students.

We discuss each of these efforts below.

2 Integrated Incremental Case-Based Understanding and Explanation: DMAP

Research Goals

We have the following goals for our research with the direct memory access algorithm for understanding and inference:

- using the DMA algorithm to carry out larger scale case-based reasoning
- improving the robustness of the understander
- exploring issues in parallelizing the algorithm

In what follows below, we will not make detailed reference to the algorithm, except to distinguish the two basic components of memory search

- concept refinement, which goes from an abstract memory structure and some components of it to the most specific version of that memory structure that contains those components, and
- concept reference, which uses concept sequences to go from references to certain component concepts to the larger concept that contains those concepts.

Concept refinement, for example, goes from "a communication event by Milton Friedman" to "Milton Friedman's argument about interest rates." Concept reference goes from "interest rates", "will" and "soar" to the concept of rising interest rates.

on For	
RA&I	<input checked="" type="checkbox"/>
B	<input type="checkbox"/>
nced	<input type="checkbox"/>
cation	
Distribution/	
Availability Code	
Dist	Avail and/or Special
A-1	

Case-based reasoning with the DMA Algorithm

Although developed for parsing, the DMA algorithm is a general process for searching memory and creating new memory structures. As such, it can be used to implement the other memory-based processes that we have studied at Yale, including reminding, failure and explanation-based generalization, and case-based reasoning. Currently, every program at Yale and elsewhere that does case-based reasoning has its own idiosyncratic model of dynamic memory and its own version of processes to search and extend that memory. In particular, most of these programs have two kinds of knowledge, represented in two very different ways:

- domain-specific facts, represented declaratively in Memory Organization Packets (MOPs) [Sch82], and
- domain-specific processes, represented procedurally in either Lisp code or IF-THEN rules.

In the DMA parser/understander (DMAP), however, all domain knowledge is represented declaratively, either in MOPs or in concept sequences attached to MOPs. Only domain-independent knowledge about searching and instantiating memory structures and recognizing concept sequences appears in Lisp code.

This unified form of knowledge representation has obvious benefits for reducing redundancy in existing systems, which sometimes have to represent the same information, *e.g.*, that the actor of a communication event must be a human, in both MOP and rule form. The unified approach also supports learning much better. From prior research, we know something about learning new MOPs through generalization and/or explanation [Sch86]. But learning new rules is equivalent to automatic programming, and is not in general a tractable problem. Since DMAP only has MOPs and concept sequences, the learning problem is much more constrained.

To be specific, one of our central research goals is to implement learning by case-based explanation in the economic reasoning domain, using DMAP. Case-based explanation, using explanation patterns (XPs) is currently being explored in the SWALE system [Kas86, LO86]. The focus of that research is on generating creative explanations for highly anomalous situations. The problem the program faces is one where no normal explanation holds, so it must explore the larger space of less obvious possibilities.

The focus of our research will be on generating plausible explanations for failures in understanding economic reasoning. When reasoning about things like the effects of interest rates on the trade deficit, or of the stock market crash on employment, the problem is that there are too many explanations, some of them inconsistent with others. The first step in making sense of economic arguments is to organize them into larger patterns, such as monetarist versus supply-side arguments. These patterns abstract out the basic nature of the different individual arguments. With these patterns made explicit, a new argument can be

more easily understood, not by trying to unravel the exact causality of the argument, which is often vaguely or even incorrectly specified, but by recognizing what basic abstract pattern the argument best fits. Cues to these patterns include not only the elements of the argument itself, but also who is giving it, and in what context. Thus, knowing that Milton Friedman is giving an argument mentioning the money supply, in a debate with Thurow in *Newsweek*, is enough to cause the recognition of Friedman's basic monetarist explanation pattern.

Recognizing the basic explanation pattern involved can then heuristically improve the system's ability to evaluate the new argument. If the system already has classified the basic monetarist position as plausible, but probably incomplete, it can likewise, without extensive causal simulation, guess that the new argument is plausible but not the whole story. Note that the "depth" to which the system understands the new argument will follow directly from how detailed and accurate the existing explanation pattern is. In this way, a DMA-based case-based reasoner using explanation patterns can be used to model both novice and expert understanding, by varying the detail of the XPs stored.

Making the Parser more Robust

The DMA approach to parsing and understanding has achieved several of our original goals. It has removed the memory-parser bottleneck, so that much more knowledge can be used when understanding text than ever before. It has become the basis for a general scheme for representing inference processes, as described above. It has inspired a new theoretical model of understanding as a memory search rather than meaning construction. It handles lexical ambiguity, the most common phenomenon ignored by almost every parsing system, smoothly and directly.

However, one of our major criticisms of other language understanding approaches still applies to the DMA system: fragility in the face of input that is either ill-formed or unexpected. Though there is certainly a difference between bad input and missing knowledge, we think that the same solutions will apply in both cases.

The core of our approach will be in extending the system's ability to know when it's in trouble. Currently, the understander has MOPs and concept sequences for recognizing when new inputs fail to match existing structures and for repairing these failures. For example, if the system reads "Joe Blow says that liberal monetary policies are driving up interest rates," it recognizes that this matches an argument given by Milton Friedman. The mismatch in who is giving the argument is reconciled by a structure that hypothesizes that Joe Blow is a monetarist like Milton Friedman. In this way, limited forms of missing knowledge can be filled in, as long as the concept sequences recognize the relevant memory structures. That is, some problems in concept refinement can be handled, as long as concept reference is correct.

Extending this approach means handling cases when problems arise with

concept reference, either because the text uses some form that the parser doesn't have a sequence for, or because the text is simply ill-formed. The DMA system needs to monitor how well the concept sequences are doing. This means we need a good characterization of what normal, successful activation of concept sequences looks like, as well as what things look like when there are problems. For example, the system can tell when an unknown word has been read because no concept sequences are advanced. It is harder to know that words are being used in unknown ways, because concept sequences will be begun or advanced as if things are OK, but later these sequences will not be able to complete, or, if they do, the concept they reference will not connect to any active concept sequence.

Besides being able to characterize when understanding problems are occurring, we will also of course need to specify what to do to correct those problems. When people have trouble understanding something, they go back and read it more carefully. DMAP, like most systems, is always reading carefully. This is appropriate for the moment, since the domain of economic argument calls for careful reading. Therefore, going back and reading more carefully doesn't make sense. It does make sense however to try reading *less carefully*. Since DMAP has already read the text carefully and had trouble, it doesn't have the knowledge necessary to understand all of the text. However, it probably can handle most of the text, if it just ignores the problematic part, and uses some default interpretation instead. To determine which parts of the text to ignore, one possibility is to have DMAP pick the best partially matching concept sequences. The research problem then becomes one of defining and implementing "best match."

Parallel DMA Understanding

Our third research goal, which is somewhat independent of the other two, is to investigate ways in which the DMA algorithm can be implemented on parallel hardware. Much of the algorithm is already inherently parallel, because it involves simple graph search. Both concept refinement to specializations of memory structures, and concept reference via multiple concept sequences can be done in parallel. However, as with all research into parallel algorithms, there are hard problems to be solved in coordinating the results of parallel processes. Furthermore, since the parallel machine we are most interested in experimenting with is the Connection Machine, there are hard implementation problems involved in representing large knowledge structures in a distributed efficient manner.

3 Opportunistic Case-based Planning Using Proverbs

Faced with a straightforward, well specified planning problem, we believe that people proceed by retrieving old standard plans which work or almost work in the new situation and modifying them to fit the new circumstances, as demonstrated in the CHEF program [Ham86]. That is, people build plans starting from cases which are known specific plans. Although this model provides a powerful account of mundane planning, as a model of all human planning it is seriously incomplete. In addition to straightforward planning problems, people frequently encounter problems which are novel or poorly specified and for which no ready-made solutions are immediately apparent.

This research focusses on case-based planning in complex competitive domains where few ready-made solutions are to be found. When the planner is unable to straightforwardly find and adapt a standard plan it falls back on cases embodying more general planning advice and attempts to adapt these to the situation at hand. The information gained from adapting these cases guides retrieval and modification of appropriate standard plans.

This research addresses three related questions:

1. **Representation:** What is the general planning knowledge these cases embody and how is it represented?
2. **Adaptation:** How is general planning knowledge adapted to specific problems?
3. **Learning:** How can a planner learn from adapting general cases to specific problems?

Representation and the role of proverbs

Consider the following thought experiment. Pick a proverb and your favorite current life crisis. Attempt to see how the proverb could be relevant to resolving the crisis. Chances are, almost regardless of which proverb you choose, you won't come away entirely empty handed. If you are lucky, you may even arrive at a genuinely new insight or a plan of action for dealing with some aspect of your problem.

We draw a moral from the human capacity to perform this kind of reasoning. Proverbs constitute a huge body of culturally shared planning advice in the form of cases. We claim that when no viable standard plan can be determined, people turn to cases expressed in a vocabulary suitable for representing the planning advice that proverbs express.

Two of our main research objectives include specifying a vocabulary for representing general planning knowledge and producing a catalog of cases represented in this vocabulary. Representing proverbs will provide a good arena for testing the power of this vocabulary.

Adaptation and system architecture

As a framework for exploring how general planning knowledge can be adapted to specific situations, we propose a theory of planning which combines a simple **plan retriever** and **plan modifier** with a powerful **adapter**. The plan retriever and modifier together form a minimal case-based planner of a conventional sort — that is, a planner whose only cases are standard plans or plan sketches. The adapter is invoked whenever the minimal planner reaches an impasse. This will happen when some piece of knowledge needed to retrieve or modify a standard plan is not immediately available. The adapter forces a fit between cases embodying general planning advice and the problem situation with the aim of making such missing knowledge available. The output of the adapter is a better specification of the goals of the planner and/or an elaborated description of the problem situation.

The adaptation process must be fully *integrated* into the planning process as a whole. By this we mean that it should be as responsive as possible to the functional requirements of the plan specifier and modifier, and it should make available to these components the information they need as early as possible. This imposes three basic constraints on the adapter. First, it must be able to *notice opportunities*: it must be able to suggest new goals or specifications of goals to the plan retriever and plan modifier, even when these emerge as a side effect of adapting a pattern for some other purpose. Second, the adapter must be *incremental*: it should only spend effort specifying goals and elaborating descriptions of the problem situation to the extent needed to further subtasks that the plan retriever and plan modifier are pursuing. Finally, the adapter must be *selective*. There are typically many ways one can attempt to adapt a general case to a specific situation. Choices must be made which tend to lead to the construction of viable plans, given the planner's current mandate and resource limitations.

In order to achieve full integration of the adapter with the rest of the planner, the *impasses* the planner encounters must be explicitly represented. If the adapter is to be able to detect unexpected opportunities to resolve existing *impasses*, it has to be able to retrieve the *impasses* from memory in appropriate circumstances and reason about the likelihood of being able to resolve them in light of the knowledge it is currently making available. Moreover, the requirements of incrementality and selectivity really amount to demanding that the adapter itself be a planner of sorts, where the goals are *impasses* of the minimal case-based planner. To be able to flexibly plan for these goals, the adapter has to be able to reason about them, which in turn requires explicit representations.

Explicitly representing *impasses* leads very naturally to the idea of building the planning system on top of an *opportunistic memory architecture*. An opportunistic memory architecture is a processing framework in which goals are indexed in terms of the knowledge structures used to plan for them, in a way that allows the goals to actively guide the system's problem solving behavior.

Here, the goals are the impasses. In its simplest incarnation, an opportunistic memory architecture maintains a queue of active goals and goes through the following basic problem solving cycle:

1. Pick the highest priority active goal and start working on it.
2. Every time a knowledge structure is acted on, check to see whether the action being performed on it could impact any of the systems' goals. This amounts to searching memory for relevant goals not under active pursuit, starting from the memory structure currently being processed. If any goals are found, then add them to the goal queue. If one of the new goals has a higher priority than the current goal being pursued, and the current goal can be suspended, then put the current goal back on the queue and start working on the new one.
3. Go to step 1.

We will implement the planner in an opportunistic memory architecture. In so doing, we will address three subsidiary research questions:

1. What exactly should count as an impasse and how should impasses be represented?
2. How should impasses be indexed in a large memory so that they can be straightforwardly detected when possible opportunities to satisfy them arise?
3. How are impasses detected?

Learning

The planner should be able to learn from its experience. A potential for learning exists whenever an impasse is resolved. At the very least, after generating an acceptable candidate plan, the system should be able to update its knowledge structures so that if the same problem were resubmitted to it no impasses would arise. That is, the minimal case-based planner would by itself be sufficient to the task generating the plan. Addressing this learning issue is a long term goal of this research.

Implementing the planner

We are currently working on an initial implementation of a planner satisfying the above description which constructs plans in the domain of terrorist crisis management. The top level goals of a terrorist crisis manager have to do with finding favorable ways to resolve specific crises and finding ways to avoid similar crises in the future. The input to the planner is a high level specification of a

set of goals and an initial description of an ongoing crisis. The output is a set of plan suggestions.

A brief transcript of the current implementation appears in the next section.

Throughout the course of this research, we will be making the following simplifying assumptions.

- Rather than trying hard to preselect cases embodying general planning knowledge by an estimate of their likely utility, the planner will attempt to adapt a wide range of cases to each problem situation. In any given instance, those which don't pan out quickly will be abandoned.
- Instead of trying to generate complete workable plans, the planner will generate large numbers of plan sketches, only a certain proportion of which will seem both interesting and plausible.

Conclusion

The goal of this research is to push forward the theory of case based planning to the point where it can account for constructing plausible plans of action in situations where existing case-based planners would fail because no standard solution seems available. The basic hypothesis is that a case-based planner solves these more difficult planning problems by adapting cases embodying general planning knowledge to the problem.

As part of working towards this goal, we hope to further our understanding of integrated processing and learning in an opportunistic memory.

4 Case-based Planning Transcript

The following two pages are a transcript of the case-based planning program operating in the domain of terrorist crisis management.

5 Publications during the Current Period

Refereed Papers

- Birnbaum, L. Let's put the AI back in NLP. In Proceedings of the Third TINLAP. Las Cruces, NM. 1987.
- Birnbaum, L. Inferential memory and linguistic creativity. Journal of Metaphor and Symbolic Activity. To appear.
- Farrell, R. Intelligent Case Selection and Presentation. Proceedings of the 11th International Joint Conference on Artificial Intelligence. Milan, Italy, August, 1987.
- Ram, A. AQUA: Asking Questions and Understanding Answers. Proceedings of the 6th AAAI. Seattle, WA. July, 1987.
- Schank, R.C. The Current State Of AI: One Man's Opinion. Scientific Datalink Microfiche Collection of Yale AI Technical Reports. AI Magazine, 8(4), Winter, 1987.
- Schank, R.C., Collins, G. and Hunter, L. Transcending Inductive Category Formation in Learning. Behavioral and Brain Sciences, 9(4), 1986.
- Schank, R.C., and Kass, A. Representing meaning in Man and Machine, To appear in *Versus*.
- Schank, R.C., and Kass, A. Natural Language Processing, What's really involved? In Proceedings of the Third TINLAP. Las Cruces, NM. 1987.
- Slade, S. The Yale Artificial Intelligence Project: A Brief History. Scientific Datalink Microfiche Collection of Yale AI Technical Reports. AI Magazine, 8(4), Winter, 1987.

Technical Reports

- Bain, W. Case-base Reasoning: A Computer Model of Subjective Assessment. Report 470, Yale Department of Computer Science, 1986.
- Birnbaum, L. Integrated processing in planning and understanding. Report 489, Yale Department of Computer Science, 1986.
- Hammond, K.J. Case-based Planning: An Integrated Theory of Planning, Learning, and Memory. Report 488, Yale Department of Computer Science, 1986.
- Hovy, E.H. Generating Natural Language Under Pragmatic Constraints. Report 521, Yale Department of Computer Science, May 1987.

- Kass, A., and Leake, D. Types of Explanations. Report 523, Yale Department of Computer Science, March 1987.
- Schank, R.C., and Farrell, R. Creativity in Education: A Standard for Computer-based Teaching. Report 518, Yale Department of Computer Science, January, 1987.
- Schank, R.C., and Owens, C. Ten Problems in Artificial Intelligence. Report 514, Yale Department of Computer Science, January, 1987.

Presentations by Roger Schank

- Mechanical Creativity. Keynote speaker. Artificial Intelligence Society of New England. Yale University, New Haven, CT. October 31, 1986.
- Six Fundamental Problems in AI. Invited speaker. Knowledge and Data, IFIP. Portugal. November 3-7, 1986.
- Keynote speaker. The First AI Congress. Melbourne, Australia. November 17-21, 1986.
- Participant. DARPA Workshop on Case-based Reasoning. Georgia Institute of Technology, Atlanta, GA. December 2-3, 1986.
- The 10 Fundamental Problems in AI. Invited speaker. Emory University, Atlanta, GA. December 4, 1986.
- Invited Panel. Theoretical Issues in Natural Language Processing (TIN-LAP). New Mexico State University, Las Cruces, NM. January 6-9, 1987.
- Mechanical Creativity. Keynote speaker. Museum of Science, Boston, MA. January 29, 1987.
- Mechanical Creativity. Invited speaker. Carnegie-Mellon University, Pittsburgh, PA. February 4, 1987.
- Third AI Satellite Symposium, An AI Productivity Roundtable. Invited panel. Texas Instruments, Dallas, TX. March 10, 1987 (first taping). April 8, 1987 (live broadcast).
- Active Learning and Case-based Reasoning. Invited speaker. International Symposium on Culture, Computer Science, and School System. AULA Foundation, Barcelona, Spain. April 1-4, 1987.
- Ten Problems in AI. Invited speaker. Ohio State University, Columbus, OH. April 24, 1987.
- Ten Problems in AI. Invited speaker. New York University, New York, NY. April 30, 1987.

- Invited panel. Theoretical Issues in Conceptual Information Processing. University of Maryland, College Park, MD. June 4-5, 1987.
- Invited speaker. Equitable Forum. New York, NY. June 26, 1987.
- Invited speaker. McKinsey Group. Princeton, NJ. July 8, 1987.
- Where AI is going and why it never got there. Invited speaker. Technologic Partners. Cambridge, MA. September 17, 1987.

Presentations by Other Participants

- Birnbaum, L. Integrated models of explanatory inference in story understanding. Invited speaker. New York University, New York, NY. November, 1986.
- Birnbaum, L. Artificial Intelligence and the functional view of the mind. Killeen Chair lecture. St. Norbert College, De Pere, WI. February, 1987.
- Birnbaum, L. A gentle attack on the role of logic in AI. Invited commentator. MIT workshop on the Foundations of Artificial Intelligence. Dedham, MA. June 1987.
- Birnbaum, L., Riesbeck, C., and Slade, S. Invited participants. DARPA Workshop on spoken language processing. University of Pennsylvania, Philadelphia, PA. March 1987.
- Ram, A. Questions and Explanation. Artificial Intelligence Society of New England (AISNE). New Haven, CT. October, 1986.
- Ram, A. Questions and Explanation. Yale Computer Science Department Liaison Program. New Haven, CT. November, 1986.
- Slade, S. The Future of AI: Learning from Experience. Task Force Meeting: Future and Impacts of Artificial Intelligence. University of Vienna, Vienna, Austria. August 21-23, 1987.

Books

- Charniak, E., Riesbeck, C., McDermott, D., and Meehan, J. *Artificial Intelligence Programming. Second Edition*. Lawrence Erlbaum Associates, 1987.
- Schank, R. *Explanation Patterns: Understanding Mechanically and Creatively*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
- Schank, R.C. *The Creative Attitude: Learning to Ask and Answer the Right Questions*. with P. Childers. McMillan and Company, New York, NY, in press.

- Schank, R. and Riesbeck, C. *Inside Computer Learning*, Lawrence Erlbaum Associates, in preparation.
- Slade, S. *The T Programming Language: A Dialect of LISP*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

Sections of Books

- Anderson, J.R., Boyle, C.F., Farrell, R., and Reiser, B.J. Cognitive Principles in the Design of Computer Tutors, in *Modelling Cognition*, P. Morris (ed.), 1987.
- Schank, R.C. and Farrell, R. Creativity in Education: A Standard for Computer-based Teaching, in *Machine-mediated Learning*, Friedman, E., and Resnikoff, H. (eds.), Taylor-Francis: New York, NY, 1987.
- Schank, R.C., and Farrell, R. Memory, in *Introduction to Cognitive Science*, McTear, M.F. (ed), Ellis Horwood, 1987.
Schank, R.C., and Kass, A. Explanations, Machine Learning, and Creativity, in *Machine Learning III*, forthcoming.
- Schank, R.C., and Owens, C. Understanding by Explaining Expectation Failures, in *Communication Failures in Discourse*, R. Reilly (ed.), North-Holland, 1987.
- Schank, R.C., and Owens, C. The Mechanics of Creativity, in *The Age of Intelligent Machines*, R. Kurzweil (ed), MIT Press, forthcoming.
- Schank, R.C., and Ram, A. Explanation and Creativity, in *Advances in Cognitive Science*, N. Sharkey (ed.), forthcoming.
- Schank, R.C., and Slade, S. Social and Economic Impacts of Artificial Intelligence, in *Impacts of Artificial Intelligence*, R. Trappl (ed.), Amsterdam: North-Holland, 1986.
- Schank, R.C., and Slade, S. The Future of Artificial Intelligence: Learning from Experience. In *Future and Impacts of Artificial Intelligence*, R. Trappl (ed.), Forthcoming.

6 References

- [Ham86] K.J. Hammond. *Case-based Planning: An Integrated Theory of Planning, Learning and Memory*. PhD thesis, Yale University, 1986. Technical Report 488.
- [Kas86] A. Kass. Modifying explanations to understand stories. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Cognitive Science Society, Amherst, MA, August 1986.
- [LO86] D. Leake and C. Owens. Organizing memory for explanation. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Cognitive Science Society, Amherst, MA, August 1986.
- [Sch82] R.C. Schank. *Dynamic memory: A theory of learning in computers and people*. Cambridge University Press, 1982.
- [Sch86] R.C. Schank. *Explanation Patterns: Understanding Mechanically and Creatively*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.